

SDM 241

大规模软件开发:过程与研发管理

朱敏博士  
微软有限公司

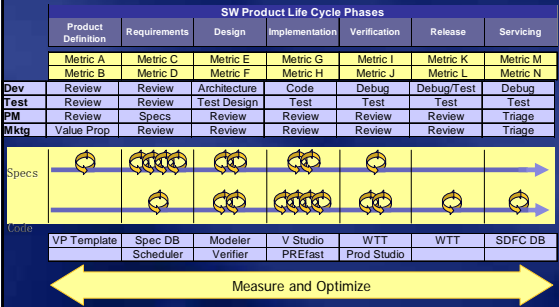
成功软件企业的特征

- 没有一个企业的成功能超越其Vision  
“Microsoft is the most successful business ever since there is business”
- 信息工作者的管理
- 合理的奖励制度 (Rewarding Structure) 是软件企业良性循环的基础
  - 短期, 中期, 长期
  - 不断更新的挑战
- 企业管理的哲学和企业文化
  - 沟通是企业运营的核心
  - 文化管理是最高层次的管理

软件开发趋势

- 激烈的竞争和快速的节奏
  - 质量/安全/方便/合成
- 产品更新换代周期减短
- 创新和执行 (Innovation meets Execution)  
“A lion leading an army of sheep is more fearsome than a sheep leading an army of lions”

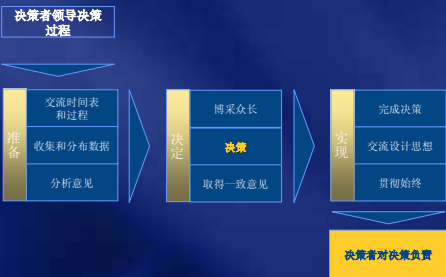
软件产品生命周期



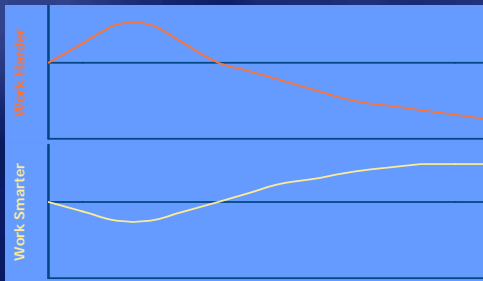
软件工程的目的

- 整合Waterfall/CMM/XP/RUP/Scrum的优势和特点
- 适应于具体实践
  - 项目规模
  - 团队大小
  - 产品或服务期限
- 最后的成功高于一切 (Success is more important than novelty)

软件过程决策



## 软件系统的生产力



## 产品质量的各个方面

- 工程设计
  - 一致性
  - 可靠性
  - 性能
  - 可维护性
  - 可推广性
  - 规模可扩展性
- 用户界面
  - 方便
  - 效用
  - 吸引力
  - 满意程度

## 研发流程

- 研发规划
- 设计过程
- 实现
- 合成
- 测试
- 发行 (RTM/RTW)

## 研发规划 1

- 研发团队参与功能需求分析
  - 需求数据库自动处理
  - MVP用户体验
  - 项目经理研究成果
- 关键与侧重点
  - 向前与向后兼容
  - 目标与非目标
  - 研发总体复杂性估计

## 研发规划 2

- 调查研究的具体领域
- 优先, 重点和风险估计
- 体系结构设计
  - 功能细分
  - 内部依赖性
  - 外部依赖性
  - 体系图解和工作流

## 设计过程

- 精确估计
  - 24\*7 法则
  - 没有重复或遗漏
  - 缓冲时间
- 详细结构
  - UML/XSD极其流水操作工具
  - 公共工具和基础类型
  - 功能集合的构成要素

## 设计模式

- 同样适用于应用软件和网络开发
- 基础设计模式
  - Abstract factory
  - Adaptor
  - Bridge
  - Façade
  - Singleton
  - Subject-observer
  - Visitor

## 面向对象的设计

- 面向对象的基本要求
  - 模块化
  - 可重用
  - 可推广
  - Inversion of control
- 基于对象的模块化和分布计算
  - CORBA, DCOM, Java RMI

## 设计框架

- 分类:
  - 系统结构 System infrastructure frameworks
  - 中间件结构 Middleware integration frameworks
  - 企业应用 Enterprise application frameworks
- 框架元素:
  - 模式 Patterns
  - 类型库 Class libraries
  - 插件 Components

## 设计原理: KISS(简单实用)

- 体系结构, 研发过程始终短小精悍
- 关于架构设计:
  - 所有设计力求简单, 但不过于简化 (Simple but not simpler).
  - 从基本结构到多次优化.
  - 系统易于理解和维护, 必要的文档说明.
  - 优美的设计通常简单而不繁琐.
  - 不等于快速草率 (Not quick and dirty).

## 实现: 编程

- 数据结构 + 算法
- 初级和高级视窗编程
- 丰富用户界面 Rich client user interface
  - Commands
  - Undo-redo
  - Drag-drop
  - Copy-paste
  - Windowing structure
  - Persistence
  - Authoring support
- 网络应用 Web Application programming
  - 多层次结构 Multi-Tier Structure
  - 面向服务的架构 SOA: Service-Oriented Architecture

## 实现: 标准

- 工程手册
  - 编程标准
  - 风格, 格式和可读性
- 程序评估
  - 任何一行程序都需评估
  - Review round-trip
  - 最后评估 (Sign-off)
- 源代码扫描评估
- 效率评估工具

实现:测试

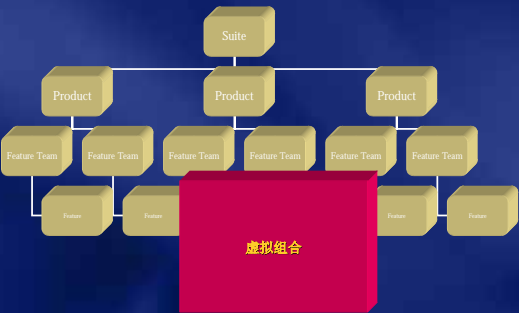
- 单元测试
  - 单元测试框架
  - 与scripting的合成
- CIT
- 虚拟团队
  - PM/DEV/Test
  - 进入和退出的规则
  - 迭代

PM/DEV/TEST



PM Spec[ 3D, TDS ]: one centralized specification.

虚拟组合



| Rank | Company                               | Revenues (\$ millions) |
|------|---------------------------------------|------------------------|
| 1    | <a href="#">Microsoft</a>             | 36,835                 |
| 2    | <a href="#">Oracle</a>                | 10,156                 |
| 3    | <a href="#">Computer Assoc. Intl.</a> | 3,354                  |
| 4    | <a href="#">Electronic Arts</a>       | 2,957                  |
| 5    | <a href="#">Veritas Software</a>      | 2,042                  |
| 6    | <a href="#">Symantec</a>              | 1,870                  |
| 7    | <a href="#">Intuit</a>                | 1,868                  |
| 8    | <a href="#">Adobe Systems</a>         | 1,667                  |
| 9    | <a href="#">BMC Software</a>          | 1,419                  |
| 10   | <a href="#">Siebel Systems</a>        | 1,340                  |

Questions and Answers

**Microsoft®**  
您的潜力. 我们的动力